Сравнение точности и скорости различных методов оценки линейной регрессии

Дергачёв Кирилл Олегович
аспирант, Московский финансово-промышленный университет «Синергия», kirill1062@yandex.ru

Данная статья посвящена сравнению методов оценки коэффициентов линейной регрессии. Для сравнения были выбраны стандартные методы линейной алгебры: напрямую используя псевдообратную матрицу Пенроуза-Мура, LDL разложение, разложение Холецкого, QR, SVD; а также методы градиентного спуска: L-BFGS-B и метод сопряжённых градиентов. Для проверки результатов в разных условиях был сгенерирован набор синтетических данных, различающихся размером, шумом наблюдаемых значений по отношению к реальной зависимости, числом обусловленности и прочими особенностями, потенциально вносящими проблемы с точностью коэффициентов или точностью тестов. Генерация данных, оценка коэффициентов и тестирование проводились на языке Python. Все используемые методы являются обёртками оптимизированных библиотек написанных на низкоуровневых языках, поэтому проведённые тесты должны быть достаточно честными в плане эффективности реализации самих методов.

Ключевые слова: Линейная регрессия, симуляция, синтетические данные.

Введение

Линейная регрессия является одним из наиболее широко известных инструментов статистического моделирования и машинного обучения, а задача оценки коэффициентов методом наименьших квадратов (МНК) — одной из фундаментальных вычислительных задач. В общем случае задача оценки коэффициентов линейной регрессии достаточно проста, но существует значительное число условий, без которых нет гарантии получить качественный ответ. Проблемы могут быть вызваны как теоретическим несоответствием базовой спецификации модели реальным данным (в таком случае даже идеальное с практической точки зрения решение даст плохой результат), так и практическими проблемами (основной пример — численная точность компьютера). В данной работе под точностью предполагается отсутствие или минимальный эффект от проблем второго типа, проблемы первого решаются изменением спецификации модели. На практике обычно рассматривают несколько алгебраических подходов: решение через нормальные уравнения с обращением матрицы X^TX, разложение Холецкого, LDL-разложение, QR-разложение и сингулярное разложение (SVD). Помимо этого как и любую другую задачу оптимизации её можно решить численно, в данной статье были выбраны два самых распространённых численный метода градиентного спуска - L-BFGS-B и метод сопряжённых градиентов (CG). Каждый из этих методов обладает своими характеристиками по скорости, потреблению памяти и устойчивости к плохой обусловленности.

Теоретически все перечисленные методы должны давать одно и то же решение — вектор коэффициентов, минимизирующий сумму квадратов остатков. Однако на практике, при работе с числами с плавающей запятой, методы могут расходиться в результатах, особенно когда матрица признаков X плохо обусловлена. Обусловленность матрицы, измеряемая числом обусловленности $\kappa(X) = \sigma_{max}/\sigma_{min}$, является ключевым фактором, определяющим чувствительность решения к малым изменениям входных данных. Методы, работающие непосредственно с матрицей X (QR, SVD), в теории сохраняют точность, определяемую $\kappa(X)$, тогда как методы, использующие X^TX (нормальные уравнения, Холецкий), сталкиваются с квадратичным ухудшением обусловленности, поскольку $\kappa(X^TX) = \kappa(X)^2$.

Настоящая работа посвящена систематическому сравнению методов решения задачи линейной регрессии с точки зрения скорости вычислений и точности восстановления истинных коэффициентов. В данной работе рассматриваются не только классические алгебраические методы (нормальные уравнения, Холецкий, LDL, QR, SVD), но и итеративный подход, в частности метод градиентного спуска и его квазиньютоновские варианты (L-BFGS-B и CG), реализованные в библиотеке SciPy. Поведение этих методов исследуется на синтетических данных с контролируемым числом обусловленности, что позволяет точно определить пороги, при которых каждый метод начинает демонстрировать потерю точности или полный отказ.

Для проведения экспериментов ранее был разработан генератор данных, позволяющий независимо контролировать размер выборки, число признаков, уровень шума, степень автокорреляции ошибок и целевое число обусловленности матрицы X. Это даёт возможность изолированно изучать влияние каждого фактора на поведение методов. В тестах измеряется вычислительное время, ошибка восстановления истинных коэффициентов, а также сумма квадратов остатков прогнозов моделей.

Материалы и методы исследований

Постановка задачи

Как было упомянуто ранее в данной работе сравнивается множество методов оценки коэффициентов линейной регрессии, как алгебраические, так и численные. Далее следует краткая формулировка решаемой проблемы и методов её решения.

В терминах линейной алгебры задача линейной выглядит как нахождение самого лучшего вектора β для решения уравнения вида:

$$X\beta = Y \quad \text{(Уравнение 1)}$$

где X - матрица размерности $n \times k$ независимых переменных (данные используемые для объяснения/прогноза Y), β - вектор длины k искомых коэффициентов, а Y - вектор длины n интересующей нас переменной, которую нужно прогнозировать или оценить через известные X. В данной системе $n > k$, поэтому точного решения не существует. Под «самого лучшего» в стандартном случае подразумевают минимизирующего Евклидово расстояние между полученными моделью и реальными значениями Y. Все методы используемые в данной статье дают именно такую оценку β.

Методы оценки коэффициентов

В рамках данной работы было проведено сравнение различных методов решающих данную проблему:

Псевдообратная матрица:

Самое очевидное решение подобной системы — решение через нормальные уравнения. В случае если матрица независимых переменных имеет полный ранг (что обычно истинно на практике), обе части уравнения можно домножить на

X^T, что позволит вывести β обратив матрицу, полученную с левой стороны уравнения:

$X\beta = Y$ (Уравнение 2a)
$X^TX\beta = X^TY$ (Уравнение 2b)
$\beta = (X^TX)^{-1}X^TY$ (Уравнение 2c)

Разложение Холецкого:

Данный метод раскладывает X^TX на произведение нижнетреугольной и верхнетреугольной матрицы, а затем решает их как 2 системы уравнений (предполагается что X^TX является квадратной положительно-определённой, что математически всегда истинно, но в некоторых случаях может не выполняться на практике из-за численной ошибки):

$X\beta = Y$ (Уравнение 3a)
$LL^T\beta = Y$ (Уравнение 3b)
$Lz = Y \rightarrow L^T\beta = z$ (Уравнения 3c)

LDL разложение:

Этот метод практически идентичен разложению Холецкого, но также выделяет диагональную матрицу D, за счёт чего не ломается если X^TX оказывается не положительно определённой, но из-за выделения этой матрицы и необходимости решить ещё одну систему уравнений может оказаться медленнее:

$X\beta = Y$ (Уравнение 4a)
$LDL^T\beta = Y$ (Уравнение 4b)
$Lz = Y \rightarrow Dw = z \rightarrow L^T\beta = w$ (Уравнения 4c)

QR разложение:

Для этого метода (и последующих) вычисление X^TX не требуется. Матрица X раскладывается на ортогональную матрицу Q и верхне-треугольную R, далее обращением Q, которое для ортогональных матриц равнозначно простому транспонированию, уравнение приводится в упрощённый вид и решается обратной подстановкой:

$X\beta = Y$ (Уравнение 5a)
$QR\beta = Y$(Уравнение 5b)
$R\beta = Q^TY = w$ (Уравнение 5c)

SVD разложение:

Матрица X раскладывается на $U\Sigma V^T$, где $U_{n\times n}$ - ортогональная матрица, $\Sigma_{n\times k}$ - диагональная матрица, $V_{k\times k}$ - другая ортогональная матрица. Преобразовав полученное выражение получаем обычную систему уравнений, решаем её и получаем ответ:

$X\beta = Y$ (Уравнение 6a)
$U\Sigma V^T\beta = Y$ (Уравнение 6b)
$\Sigma V^T\beta = U^TY$ (Уравнение 6c)
$\Sigma z = U^TY \rightarrow \beta = Vz$ (Уравнения 6d)

Библиотека LAPACK [5][6], используемая для вычисления данных матриц, содержит 2 варианта разложения более старый использующий итерации QR разложений, и более новый рекурсивный метод.

Градиентный спуск (метод сопряжённых градиентов):

Методы градиентного спуска итеративно двигаются в направлении противоположном градиенту (в задачах минимизации) пока не достигнут минимума по заранее заданной метрике (например, изменение функции потерь меньше заданного значения) или не пробьют ограничение по количеству шагов. Общая идея подобных методов:

$X\beta = Y$ (Уравнение 7a)
$(X\beta - Y)^T(X\beta - Y) \rightarrow \min\limits_{\beta}$ (Уравнение 7b)
$\beta^TX^TX\beta - 2Y^TX\beta + Y^TY \rightarrow \min\limits_{\beta}$ (Уравнение 7c)
$\nabla f(\beta) = 2(X^TX\beta - X^TY)$ (Уравнение 7d)
$\beta_{i+1} = \beta_i - \alpha_i\nabla f(\beta_i)$ (Уравнение 7e)

Выбирается какое-то первое приближение искомого вектора β_0, далее оно итеративно обновляется двигаясь обратно градиенту (α_i - коэффициент определяющий масштаб шага).

Проблема данного метода в том что зачастую каждый шаг оптимален локально, но не оптимален глобально и поэтому итерации могут скакать вокруг оптимума.

Метод сопряжённых градиентов решает эту проблему выбирая вектора сопряжённые по гессиану функции и двигаясь по полученному новому «базису».

Градиентный спуск L-BFGS-B:

Если на каждом шаге учитывать вторую производную (гессиан), то за счёт более точной аппроксимации сходимость метода градиентного спуска будет лучше. При известном гессиане шаг спуска выглядит как:

$\beta_{i+1} = \beta_i - H(\beta_i)^{-1}\nabla f(\beta_i)$ (Уравнение 8)

Проблема данного метода — необходимость считать гессиан на каждом шаге. Это затратно как в плане времени необходимого для вычислений, так и по памяти. Также в некоторых случаях для второй производной нет явной формулы. BFGS (алгоритм Бройдена — Флетчера — Гольдфарба — Шанно) решает эту проблему выбирая фиксированный стартовый гессиан и обновляя его на каждом шаге используя производные. Полученная матрица является достаточно хорошим приближением, считается относительно быстро и для данного метода можно сократить потребление памяти, используя только информацию из недавних шагов (вся матрица в памяти не хранится).

Изменение в генерации данных

В ходе генерации данных было выявлено что реализованный метод для генерации независимых переменных с большим числом обусловленности не даёт достаточно «плохую» задачу чтобы значительно повлиять на работу алгоритмов, поэтому был добавлен новый метод, и изменён интерфейс для возможности выбора числа обусловленности напрямую. Данный метод генерирует матрицу независимых переменных используя определение числа обусловленности напрямую. Сначала случайно генерируются ортогональные матрицы SVD разложения а затем значения в диагональной генерируются по логарифмической шкале от 1 до необходимого числа обусловленности. Перемножив эти матрицы получается искомая.

Используемые инструменты

Практическая часть данной работы была выполнена на языке Python. Данные для оценки являются синтетическими, и получены при помощи методов описанных в [8]. Весь код реализован на языке Python, версия 3.13.5. Найти его можно [9]. Зависимости и их версии зафиксированы в файле pyproject.toml. Все тесты были выполнены на домашнем ПК (ЦП AMD Ryzen 9 5900X, 32Гб DDR4 ОЗУ + 15Гб swap-раздел на m.2 диске).

Параметры генерации данных

Для генерации данных были выбраны стандартизированные параметры с интуитивным объяснением для более простой интерпретации и краткого описания. Названия датасетов построены по формату: <размер датасета>_<качество X>_<масштаб коэффициентов>_<качество Y>. Далее описание этих параметров с их короткими названиями:

размер датасета (размерность матрицы независимых переменных): large (75000 * 3000), small (100 * 5)

качество X: gx (дисперсия при генерации независимых переменных равная 3), bx (число обусловленности равное 10^{10})

масштаб коэффициентов (мат. ожидание коэффициентов): sc (5), lc (10^7). Параметр разброса значений и доля отрицательных в обоих случаях 0.5.

Качество Y (автокорреляция): gy (0), by (0.4). Для обоих случаев выставлена дисперсия 20^2

Из всех возможных комбинаций данных параметров собираются условия генерации данных. Далее для краткости будут использоватся эти названия (пример: «large_gx_sc_gy»).

Литературный обзор

Задача оценки параметров линейной регрессии методом наименьших квадратов (МНК) имеет долгую историю, восходящую к работам Лежандра и Гаусса начала XIX века, и остаётся одной из фундаментальных вычислительных задач в статистике и машинном обучении [1]. Современные приложения МНК варьируются от обработки изображений и геофизики до финансового моделирования и анализа климатических данных, что делает вопросы выбора эффективного и устойчивого вычислительного метода по-прежнему актуальными [2].

Классические подходы к решению переопределённых систем $X\beta = Y$ делятся на две большие группы: методы, работающие непосредственно с матрицей признаков X, и методы, использующие матрицу нормальных уравнений X^TX. Первая группа включает QR-разложение и сингулярное разложение (SVD), вторая — решение через нормальные уравнения, разложение Холецкого и LDL-разложение. Фундаментальное различие между ними состоит в том, как они реагируют на плохую обусловленность. Как отмечается в работе [3], точность решения через нормальные уравнения зависит от квадрата числа обусловленности $\kappa(X)^2$, тогда как QR и SVD сохраняют точность, определяемую непосредственно $\kappa(X)$. Это делает методы, работающие с X^TX, предпочтительными только для хорошо обусловленных задач [4].

Помимо прямых алгебраических методов, задача МНК может решаться итеративно как задача оптимизации. Простейший подход — градиентный спуск — двигается в направлении, противоположном градиенту функции потерь, однако на плохо обусловленных задачах он сходится крайне медленно, поскольку каждый шаг оптимален лишь локально. Метод сопряжённых градиентов (CG) решает эту проблему, выбирая направления поиска, сопряжённые по матрице X^TX, что позволяет достичь точного решения за не более чем n шагов и даёт скорость сходимости, зависящую от $\kappa(X)$, а не от $\kappa(X)^2$ [7]. Квазиньютоновские методы, в частности BFGS и его модификация с ограниченной памятью L-BFGS, аппроксимируют обратный гессиан, используя только градиентную информацию, что позволяет достичь сверхлинейной сходимости без явного вычисления вторых производных. L-BFGS особенно привлекателен для задач с большим числом параметров, поскольку хранит лишь несколько векторов истории вместо полной матрицы $n \times n$. Для сравнения с прямыми методами в данной работе выбраны CG как представитель чисто первопорядковых методов и L-BFGS-B как наиболее распространённая и практически значимая реализация квазиньютоновского семейства.

Результаты и обсуждения

Для сравнения методов использовались две метрики: среднеквадратичная ошибка прогнозов модели (RMSE) на тестовой выборке и время на нахождение результата. Сравнение RMSE на истинных коэффициентах как основная метрика было бы некорректным: в присутствии шума, автокорреляции и гетероскедастичности лучшая по MSE прогноза оценка коэффициентов не совпадает с истинными коэффициентами модели, но она предоставлена для демонстрации масштаба.

Результаты всех тестов доступны в репозитории [9]. В таблицах ниже приведены результаты для условий, различающихся только числом обусловленности и размером датасета; остальные параметры генерации на качественную картину не влияли.

Таблица 1.
Маленький датасет, небольшое число обусловленности (1.48).
(MSE на истинных коэффициентах: 20.9394)

	direct	LDL	Cholesky	QR	SVD(DC)	SVD(QR)	BFGS	CG
RMSE	20.8368							
Время (с.)	0.00029	9.1e-5	0.00025	9.4e-5	0.00011	9.3e-5	0.00144	0.00073

Таблица 2
Маленький датасет, большое число обусловленности (9999998638).
(MSE на истинных коэффициентах: 20.4603)

	direct	LDL	Cholesky	QR	SVD(DC)	SVD(QR)	BFGS	CG
RMSE	20.232	20.225	-	20.218	20.218	20.218	20.305	20.236
Время (с.)	0.0001	9.3e-5	-	5.5e-5	9.7e-5	7.3e-5	0.0017	0.0027

Таблица 3.
Большой датасет, небольшое число обусловленности (1.5).
(MSE на истинных коэффициентах: 175.4618)

	direct	LDL	Cholesky	QR	SVD(DC)	SVD(QR)	BFGS	CG
MSE	163.297							
Время (с.)	8.31	3.20	3.51	76.25	15.85	57.99	4.19	9.8

Таблица 4.
Большой датасет, большое число обусловленности (9999999986).
(MSE на истинных коэффициентах: 20.069)

	direct	LDL	Cholesky	QR	SVD(DC)	SVD(QR)	BFGS	CG
MSE	27.398	20.386	-	19.67	19.67	19.67	-	-
Время (с.)	7.198	2.463	-	79.513	16.19	36.978	-	-

Малые датасеты, хорошая обусловленность (Таблица 1): Все методы дают практически идентичные значения RMSE - различия лежат в пределах 12-го знака. Выбор метода определяется исключительно скоростью. Наименьшее время показали LDL и QR, за ними SVD и Холецкий. Градиентные методы оказались на порядок медленнее при той же точности.

Малые датасеты, плохая обусловленность (Таблица 2): Картина качественно меняется. Разложение Холецкого во всех прогонах завершается ошибкой “singular matrix” - матрица X^TX теряет положительную определённость из-за накопленных ошибок округления. Прямое решение через нормальные уравнения даёт качество катастрофически далекое от ожидаемого. LDL оказывается устойчивее: RMSE ≈ 20.22, сравнимый с QR и SVD. QR, SVD(DC) и SVD(QR) дают наилучший RMSE (20.218).

Большие датасеты, хорошая обусловленность (Таблица 3): Точность всех методов одинакова, различие — только во времени. Методы на основе X^TX заметно быстрее. Градиентные методы демонстрируют большой разброс: от 4.2 с (BFGS в одном из прогонов) до 126 с (CG), что отражает сильную зависимость их сходимости от конкретной реализации генератора. Это соответствует теоретическим оценкам сложности: $O(nk^2)$ для методов на основе X^TX, примерно вдвое больше для QR и в несколько раз больше для SVD.

Большие датасеты, плохая обусловленность (Таблица 4): Как и на малых датасетах, разложение Холецкого падает с ошибкой “singular matrix”. Прямое решение через нормальные уравнения снова даёт большое RMSE - прогнозы непригодны. LDL устойчивее. QR, SVD(DC) и SVD(QR) дают наилучший RMSE (19.67). Градиентные методы (CG и L-BFGS-B) на этих задачах не уложились в отведённый лимит времени (15 минут) и были остановлены — это принципиальное ограничение итеративных методов при больших размерах задачи и плохой обусловленности.

Влияние остальных параметров: Масштаб коэффициентов и автокорреляция ошибок не меняют качественной картины: те же методы выигрывают и проигрывают, абсолютные значения метрик сдвигаются, но относительное расположение методов сохраняется. Это подтверждает, что определяющими факторами являются число обусловленности и размер датасета.

Выводы

Для хорошо обусловленных задач оптимальны методы на основе X^TX: они дают ту же точность, что и QR/SVD, но заметно быстрее. Для плохо обусловленных задач эти методы либо теряют точность (прямое решение, Cholesky), либо требуют осторожной реализации (LDL). Устойчивыми остаются только QR и SVD, причём на больших датасетах SVD с divide-and-conquer оказывается в 2–5 раз быстрее QR, что делает его предпочтительным выбором для больших плохо обусловленных задач. Градиентные методы на малых плохо обусловленных задачах дают неожиданно хорошее восстановление коэффициентов, но на больших задачах не укладываются в разумное время, что делает их непригодными в качестве основного инструмента для задач такого масштаба.

Литература

1. Trefethen L. N., Bau, III D. Numerical linear algebra. – Society for industrial and applied mathematics, 1997.
2. Golub G. H., Van Loan C. F. Matrix computations. – JHU press, 2013. – Т. 3.
3. Лихт М. К. Обусловленность в задаче о минимуме квадратичного функционала //Журнал вычислительной математики и математической физики. – 1963. – Т. 3. – №. 6. – С. 979-987.
4. Шевцов Г. С., Крюкова О. Г., Мызникова Б. И. *Численные методы линейной алгебры*. — СПб.: Лань, 2022. — 496 с.
5. Anderson E. et al. LAPACK users' guide. – Society for industrial and applied mathematics, 1999.
6. Virtanen P. et al. SciPy 1.0: fundamental algorithms for scientific computing in Python //Nature methods. – 2020. – Т. 17. – №. 3. – С. 261-272.
7. Nocedal J., Wright S. Numerical Optimization. – 2006.
8. Дергачёв К.О. Генерация синтетических данных для задач линейной регрессии Modern Economy Success. 2024. № 6. С. 383 – 388. DOI: 10.58224/2500-3747-2024-6-383-388
9. Репозиторий автора, Github, США. URL: https://github.com/kdergachev/regresson_method_comparison

Comparing speed and accuracy of linear regression fitting

Dergachev K.O.
Moscow Financial and Industrial University «Synergy»

JEL classification: C01, C02, C1, C4, C5, C6, C8

This article is devoted to the comparison of methods for fitting a linear regression model. Standard linear algebra methods were chosen for evaluation: directly using the pseudo-inverse Moore-Penrose matrix, LDL decomposition, Cholesky decomposition, QR, SVD, as well as the gradient descent methods: L-BFGS-B, conjugate gradient method. To verify the results under different conditions, a set of synthetic data was generated, differing in size, noise of observed values in relation to the real dependence, condition number and other features that potentially introduce problems with the accuracy of coefficients or tests.

Keywords: linear regression, simulation, synthetic data.

References

1. Trefethen L. N., Bau, III D. Numerical linear algebra. – Society for industrial and applied mathematics, 1997.
2. Golub G. H., Van Loan C. F. Matrix computations. – JHU press, 2013. – Т. 3.
3. Лихт М. К. Обусловленность в задаче о минимуме квадратичного функционала //Журнал вычислительной математики и математической физики. – 1963. – Т. 3. – №. 6. – С. 979-987.
4. Шевцов Г. С., Крюкова О. Г., Мызникова Б. И. *Численные методы линейной алгебры*. — СПб.: Лань, 2022. — 496 с.
5. Anderson E. et al. LAPACK users' guide. – Society for industrial and applied mathematics, 1999.
6. Virtanen P. et al. SciPy 1.0: fundamental algorithms for scientific computing in Python //Nature methods. – 2020. – Т. 17. – №. 3. – С. 261-272.
7. Nocedal J., Wright S. Numerical Optimization. – 2006.
8. Dergachev K.O. Generating synthetic data for linear regression problems. Modern Economy Success. 2024. 6. P. 383 – 388. DOI: 10.58224/2500-3747-2024-6-383-388
9. Author’s repository, Github, USA. URL: https://github.com/kdergachev/regresson_method_comparison